# SYSTEM AND METHOD FOR UTILIZING A DRAG AND DROP TECHNIQUE TO COMPLETE ELECTRONIC FORMS

## Relationship to Other Applications

This application claims the benefit of U.S. Provisional Application No. 60/159,564, entitled EXTENSION TO THE LIGHT CODE SERVER USING A DRAG-AND-DROP TECHNIQUE FOR AUTOMATICALLY FILLING OUT ELECTRONIC FORMS, and filed June 16, 1999. Provisional Application No. 60/159,564 is specifically incorporated herein by reference.

## Field of the Invention

In general, the present invention relates to computer software for filling out forms, and, in particular, to a method and system for completing one or more fields of an electronic form.

## Background of the Invention

Individual computing is continuously increasing. In general, a typical computer user has access to multiple computer systems, many of which are connected to external computer networks, such as the Internet. For example, users may have access to the Internet at work, at school, at home or via a portable device. Additionally, users may utilize establishments that rent computer systems and provide Internet access on a time basis, such as by the hour.

By connecting to the Internet, a user can access a variety of information available on networks such as the Word Wide Web ("WWW" or "Web"). The presentation of the information generally ranges from static textual and graphic images to more interactive presentations requiring action and/or information from the user. Accordingly, a user may repeatedly access the same information on the Internet

from a variety of computer systems dependent on the time of day and the proposed use (e.g., business vs. personal).

As computer network use increases, the amount of information that is transmitted between computer systems within the common network also increases. For example, within a network such as the Internet, WWW browser application programs on a computer system allow a user to view and extract information from the Internet that is provided by one or more content provider computer systems. One manner of communicating information from a user computer system to the content provider requires the user to actively complete and submit one or more data fields on a Web site, generally referred to as an electronic form. Often, the fields on the form request standard information such as a user name, a user password and/or additional contact information. Additionally, with respect to electronic commerce, or e-commerce, Internet applications, the requested information can include credit information and user-specific preferences, such as sizes, color preferences, style preferences, and/or a shipping address.

Many content providers require a user to fill out an electronic form to access their information or to utilize the service being offered. If numerous Web sites require the user to fill out the same information or if a user utilizes the same Web site repeatedly, having to fill out the same information each time, the completion of electronic forms becomes a time consuming and inefficient process.

One conventional method of assisting a user in filling out an electronic form entails having the user load and utilize a client-based application program on the user's computer system to assist the user in completing the electronic form fields. In this method, the user installs the client-based application program into the user's computer system and inputs his or her personal information. The information is then stored locally on the user's computer. Accordingly, when the user is required to fill out an electronic form, the client-based application program is initiated and the fields corresponding to the user's input are completed. Some conventional client-based application programs require an affiliation with the content provider to properly complete the form. Other client-based application programs provide the user with a display of their user information, which the user selectively transfers to the electronic form, and do not require any affiliation with the electronic form provider.

Another method for assisting a user in filling out an electronic form is referred to as a "transactor" method which entails a third party computer system directly interacting with the electronic form provider. Unlike the client-based

application program, the transactor method does not require the user to install or maintain software on the user computer system. Instead, personal information items are inputted and stored in a database on a remote, third party server. To transfer the information, the user accesses the third party server by activating a connection to the

5     third party transactor system. The transactor system then establishes a communication link with the content provider and transmits the user information to the content provider.

There are, however, deficiencies associated with both the client-based and transactor methods. The client-based method requires a user to install and maintain

10     software on his or her computer system. As mentioned previously, however, a user may utilize two or more computer systems on a frequent basis. Accordingly, the client-based method would require the user to install the client-based software on each system. Additionally, the user would have to maintain user information on each computer system as well. From a security perspective, if more than one user utilizes

15     the same computer system, the user's private information may be accessible by an unauthorized person using the same computer system. Moreover, another deficiency associated with the client-based application method is that users are often restricted from installing and maintaining unauthorized (nonstandard) software on the computer systems they use. Accordingly, a user utilizing multiple computer systems

20     could be restricted as to which computer systems are allowed to run the client-based application program.

Although the transactor method attempts to overcome this deficiency by not requiring the user to install or maintain software on the user system, the transactor method requires cross-communication between application program windows. It is

25     generally known that such communication may present a security risk of the user's information. Additionally, cross-communication between two applications may be more time consuming and reduces the benefit of the method.

Based on the above-mentioned deficiencies in the related art, there is a need for a system and method for completing electronic forms that does not require a user

30     to install or maintain software on a specific user computer system and that can be utilized by a user from any number of computer systems. Additionally, there is a need for a system and method for completing electronic forms that does not require a cross-communication of the user information.

### Summary of the Invention

Described generally, a method for completing an electronic form is provided. In accordance with the method, an electronic form having one or more fields to be completed is obtained. A user profile containing user data corresponding to the one or more fields of the electronic form is also obtained. Upon the transfer of the user profile to the electronic form, at least one of the fields of the electronic form is completed with the user data.

Further described, a system and method for utilizing a drag and drop technique to complete electronic forms are provided. A document browser computer accesses one or more electronic forms provided by a document server via an active Internet browser application window. A URL link to a fill server is established and a separate application window is created on the document browser computer display. A user ID, password and signature of the electronic form are passed from the document browser computer to the fill server. The fill server creates a fill bundle embodied in shippable code corresponding to a merger of raw user data from a user profile and a form map derived from the electronic form signature. The fill bundle is returned to the document browser computer and presented on the document browser computer's display as an icon within an open fill bundle window. The fill bundle icon is selected and dragged from the fill bundle window to the electronic form application window causing the shippable code to be executed. Accordingly, one or more fields on the electronic form are completed and the form may be submitted to the requesting provider.

In accordance with other aspects of the invention, a system, a method for generating a graphical display, a method for creating a user profile, and one or more computer-readable mediums containing computer-readable instructions are provided.

Brief Description of the Drawings

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is a block diagram of a representative portion of the Internet;

FIGURE 2 is a block diagram showing an illustrative operating environment in accordance with the present invention;

FIGURE 3 is a block diagram depicting an illustrative architecture for a computer system utilized to provide an electronic document in accordance with the present invention;

FIGURE 4 is a block diagram depicting an illustrative architecture for a computer system having a WWW browser for accessing an electronic document in accordance with the present invention;

FIGURE 5 is a block diagram depicting an illustrative architecture of a computer system utilized to generate and transmit a fill bundle in accordance with the present invention;

FIGURE 6 is a flow diagram illustrative of a routine implemented by the document browser computer of FIGURE 4 for completing one or more fields of an electronic form;

FIGURE 7 is illustrative of a window produced by a WWW browser installed on the document browser computer of FIGURE 4 embodying an electronic form to be completed by a user;

FIGURE 8 is a flow diagram illustrative of a routine implemented by the fill server computer system of FIGURE 5 for registering and/or logging in a user;

FIGURE 9 is illustrative of a window produced by a WWW browser installed on the document browser computer of FIGURE 4 for registering or logging in a user;

FIGURES 10A and 10B are illustrative of a window produced by a WWW browser installed on the document browser computer of FIGURE 4 for completing an initial registration of a user;

FIGURE 11 is illustrative of a window produced by a WWW browser installed on the document browser computer of FIGURE 4 for modifying and/or expanding a user's personal information;

FIGURE 12 is illustrative of a window produced by a WWW browser installed on the document browser computer of FIGURE 4 as a registration/login for a user;

FIGURE 13 is of a window produced by a WWW browser installed on the document browser computer of FIGURE 4 presenting a user with a fill bundle icon in accordance with the present invention;

FIGURES 14A and 14B are illustrative of the interaction between the electronic form window of FIGURE 7 and the fill bundle window of FIGURE 13 in accordance with the present invention;

FIGURE 14C is illustrative of the electronic form window of FIGURE 7 after the fields have been completed in accordance with the present invention;

FIGURE 15 is a flow diagram illustrative of a routine implemented by the fill server computer system of FIGURE 5 for generating a fill bundle in accordance with the present invention;

FIGURE 16 is a flow diagram illustrative of the routine implemented by the fill server computer system of FIGURE 5 for merging an electronic form map and raw user data into a corresponding fill bundle in accordance with the present invention; and

FIGURE 17 is a flow diagram illustrative of the routine implemented by the fill server computer system of FIGURE 5 for updating a user profile in accordance with the present invention.

<div align="center">Detailed Description of the Invention</div>

As described above, aspects of the present invention are embodied in a WWW site accessible via the Internet. As is well known to those skilled in the art, the term "Internet" refers to the collection of networks and routers that use the Transmission Control Protocol/Internet Protocol ("TCP/IP") to communicate with one another. A representative section of the Internet 20 is shown in FIGURE 1, in which a plurality of local area networks ("LANs") 24 and a wide area network ("WAN") 26 are interconnected by routers 22. The routers 22 are special purpose computers used to interface one LAN or WAN to another. Communication links within the LANs may be twisted wire pair, or coaxial cable, while communication links between networks may utilize 56 Kbps analog telephone lines, 1 Mbps digital T-1 lines, 45 Mbps T-3 lines or other communications links known to those skilled in the art. Furthermore, computers and other related electronic devices can be remotely connected to either the LANs 24 or the WAN 26 via a modem and temporary telephone or wireless link. It will be appreciated that the Internet 20 comprises a vast number of such interconnected networks, computers, and routers and that only a small, representative section of the Internet 20 is shown in FIGURE 1.

The Internet has recently seen explosive growth by virtue of its ability to link computers located throughout the world. As the Internet has grown, so has the WWW. As is appreciated by those skilled in the art, the WWW is a vast collection of interconnected or "hypertext" documents written in HyperText Markup Language ("HTML"), or other markup languages, that are electronically stored at "WWW sites" or "Web sites" throughout the Internet. A WWW site is a server connected to the Internet that has mass storage facilities for storing hypertext documents and that runs administrative software for handling requests for those stored hypertext documents.

A hypertext document normally includes a number of hyperlinks, i.e., highlighted portions of text which link the document to another hypertext document possibly stored at a WWW site elsewhere on the Internet. Each hyperlink is associated with a Uniform Resource Locator ("URL") that provides the exact location of the linked

5      document on a server connected to the Internet and describes the document. Thus, whenever a hypertext document is retrieved from any WWW server, the document is considered to be retrieved from the WWW. As is known to those skilled in the art, a WWW server may also include facilities for storing and transmitting application programs, such as application programs written in the JAVA® programming

10     language from Sun Microsystems, for execution on a remote computer. Likewise, a WWW server may also include facilities for executing scripts and other application programs on the WWW server itself.

A consumer or other remote user may retrieve hypertext documents from the WWW via a WWW browser application program. A WWW browser, such as

15     Netscape's NAVIGATOR® or Microsoft's Internet Explorer, is a software application program for providing a graphical user interface to the WWW. Upon request from the consumer via the WWW browser, the WWW browser accesses and retrieves the desired hypertext document from the appropriate WWW server using the URL for the document and a protocol known as HyperText Transfer Protocol

20     ("HTTP"). HTTP is a higher-level protocol than TCP/IP and is designed specifically for the requirements of the WWW. It is used on top of TCP/IP to transfer hypertext documents between servers and clients. The WWW browser may also retrieve application programs from the WWW server, such as JAVA applets, for execution on the client computer.

25     Referring now to FIGURE 2, an actual embodiment of the present invention will be described. A user computer system having a WWW browser application connects to the Internet 20 via a modem or other communication device. The user computer system is generally referred to as the document browser computer 30. The document browser computer 30 may interact with a variety of Web sites, such as a

30     Web site provided by a document server 32 located elsewhere on the Internet 20. Because a user may utilize more than one computer system, FIGURE 2 illustrates multiple document browser computers 30, which may be remote from one another. Generally, the document browser computer 30 connects to the Internet 20 via a modem or other communication device. The document browser computer 30 may

also comprise another type of computing device such as a palm-top computer, personal digital assistant or the like.

Also connected to the Internet 20 is a fill server computer system 34. As will be explained in greater detail below, the fill server 34 is in communication with a form map server 36 having access to a form database 38 containing one or more electronic form maps. Additionally, the fill server 34 is also in communication with a user information server 40 having access to a user information database 42 containing one or more user profiles. As would be generally understood by one skilled in the relevant art, the fill server 34 may be remote from the form map server 36 and/or the user information server 40. Moreover, the combination of one or more computer systems to provide the functions of the either the fill server 34, the form map server 36 and/or the user information server 40 is considered to be within the scope of the present invention.

The present invention is directed toward the acquisition, by a document browser computer 30, of one or more electronic forms from a document server 32. Additionally, the present invention is further directed toward the generation of a fill bundle icon corresponding to the merging of user specific information with the required data fields of the electronic form. Utilizing a graphical interface, the fill server 34 generates the fill bundle icon within the WWW browser 70 of the document browser computer 30 that can be selected, dragged and deposited on a graphical window containing the electronic form to automatically complete the fields on the form corresponding the user data stored in the user information database 42.

FIGURE 3 is a block diagram depicting an illustrative architecture for a document server 32 (FIGURE 2) utilized to provide an electronic document in accordance with the present invention. Those of ordinary skill in the art will appreciate that the document server 32 includes many more components then those shown in FIGURE 3. However, it is not necessary that all of these generally conventional components be shown in order to disclose an illustrative embodiment for practicing the present invention. Moreover, although the computer system described in FIGURE 3 is described as a server, it will be appreciated that the function of the document server may be implemented by computer systems not generally classified as server-type computer systems. Further, although only one document server 32 is depicted in FIGURE 2, it will be appreciated that other document servers 32 may be located elsewhere on the Internet 20 and utilized to serve electronic forms and documents to a user's document browser computer 30.

As shown in FIGURE 3, the document server 32 includes a network interface 44 for connecting directly to a LAN or a WAN, or for connecting remotely to a LAN or WAN. Those of ordinary skill in the art will appreciate that the network interface 44 includes the necessary circuitry for such a connection, and is also

5 constructed for use with the TCP/IP protocol, the particular network configuration of the LAN or WAN it is connecting to, and a particular type of coupling medium. Additionally, the document server 32 may also be equipped with a modem 48 for connecting to the Internet 20 through a point to point protocol ("PPP") connection or a SLIP connection as known to those skilled in the art.

10 The document server 32 also includes a processing unit 46, a display 50, and a mass memory 52. The mass memory 52 generally comprises a random access memory ("RAM"), a read-only memory ("ROM") and a permanent mass storage device, such as a hard disk drive, tape drive, optical drive, floppy disk drive, or combination thereof. The memory 52 stores an operating system 56 for controlling

15 the operation of the document server 32. It will be appreciated that the operating system component 56 may comprise a general-purpose server operating system as is known to those of ordinary skill in the art, such as UNIX, LINUX™, or Microsoft WINDOWS NT®.

The memory 52 also includes a WWW browser 54, such as Netscape's

20 NAVIGATOR® or Microsoft's Internet Explorer browsers, for accessing the WWW. Additionally, the memory 52 further includes one or more electronic documents 58 which are to be provided to users on the WWW via the WWW browser 54. It will be appreciated that these components may be stored on a computer-readable medium and loaded into memory 52 of the document server 32 using a drive mechanism

25 associated with the computer-readable medium, such as a floppy, CD-ROM or DVD-ROM drive. The memory 52, network interface 44, display 50, and modem 48 are all connected to the processing 46 unit via one or more buses. As would be generally understood, other peripherals may also be connected to the processing unit in a similar manner.

30 FIGURE 4 is a block diagram depicting an illustrative architecture for a document browser computer 30 (FIGURE 2) utilized to access and manipulate an electronic document in accordance with the present invention. Those of ordinary skill in the art will appreciate that the document browser computer 30 includes many more components then those shown in FIGURE 4. However, it is not necessary that

all of these generally conventional components be shown in order to disclose an illustrative embodiment for practicing the present invention.

As shown in FIGURE 4, the document browser computer 30 includes a network interface 60 for connecting directly to a LAN or a WAN, or for connecting

5 remotely to a LAN or WAN. Those of ordinary skill in the art will appreciate that the network interface 60 includes the necessary circuitry for such a connection, and is also constructed for use with the TCP/IP protocol, the particular network configuration of the LAN or WAN it is connecting to, and a particular type of coupling medium. The document browser computer 30 may also be equipped with a

10 modem 64 for connecting to the Internet through a point to point protocol ("PPP") connection or a SLIP connection as known to those skilled in the art.

The document browser also includes a processing unit 62, a display 66, and a memory 68. The memory 68 generally comprises a random access memory ("RAM"), a read-only memory ("ROM") and a permanent mass storage device, such

15 as a disk drive. The memory 68 stores an operating system 74 for controlling the operation of the document browser computer 30. In once actual embodiment of the invention, the operating system 74 provides a graphical operating environment, such as Microsoft Corporation's WINDOWS® graphical operating system in which activated application programs are represented as one or more graphical application

20 windows with a display visible to the user. The memory 68 also includes a WWW browser 70, such as Netscape's NAVIGATOR® or Microsoft's Internet Explorer browsers, for accessing the WWW. As will be explained in greater detail below, the WWW browser 70 includes a link 72 corresponding to a URL of a fill server 34 in accordance with the present invention. It will be appreciated that these components

25 may be stored on a computer-readable medium and loaded into the memory 68 of the document browser computer 30 using a drive mechanism associated with the computer-readable medium, such as a floppy, CD-ROM or DVD-ROM drive. The memory 68, network interface 60, display 66, and modem 64 are all connected to the processing 62 unit via one or more buses. As would be generally understood, other

30 peripherals may also be connected to the processing unit in a similar manner.

FIGURE 5 is a block diagram depicting an illustrative architecture for a fill server 34 (FIGURE 2) utilized to generate and transmit a fill bundle in accordance with the present invention. Those of ordinary skill in the art will appreciate that the fill server 34 includes many more components then those shown in FIGURE 5.

35 However, it is not necessary that all of these generally conventional components be

shown in order to disclose an illustrative embodiment for practicing the present invention. As shown in FIGURE 5, the fill server 34 is connected to the Internet 20 via a network interface 76. Those of ordinary skill in the art will appreciate that the network interface 76 includes the necessary circuitry for connecting the fill server 34

5    to the Internet 20, and is constructed for use with the TCP/IP protocol.

The fill server 34 also includes a processing unit 78, a display 82, and a mass memory 84, all connected via a communication bus, or other communication device. The mass memory 84 generally comprises a RAM, ROM, and a permanent mass storage device, such as a hard disk drive, tape drive, optical drive, floppy disk drive,

10    or combination thereof. The mass memory 84 stores an operating system 86 for controlling the operation of the fill server 34. It will be appreciated that this component may comprise a general-purpose server operating system as is known to those of ordinary skill in the art, such as UNIX, LINUX™, or Microsoft WINDOWS NT®.

15    The mass memory 84 also stores the program code and data for registering and logging-in users and for generating fill bundles in accordance with the present invention. More specifically, the mass memory 44 stores a fill bundle generator application program 88 formed in accordance with the present invention for creating a fill bundle corresponding to a merger of an electronic form and a user profile. The

20    fill bundle generator application 88 comprises computer executable instructions which, when executed by the fill server 34, generates one or more WWW browser displays as will be explained in greater detail below. Mass memory 84 also stores a user registration and login application 90 for registering new users or logging registered users. The operation of the user registration and login application 90 will

25    be described in greater detail as described below. It will be appreciated that these components may be stored on a computer-readable medium and loaded into the memory 84 of the fill server 34 using a drive mechanism associated with the computer-readable medium, such as a floppy, CD-ROM or DVD-ROM drive.

Referring now to FIGURE 6, a flow diagram illustrative of a routine 600

30    implemented by the document browser computer 30 for completing one or more fields of an electronic form will be described. The routine 600 begins at block 602, where an electronic form is obtained by the Web browser 70 on the document browser computer 30. The electronic form may be downloaded directly from the Internet 20, can be recalled from a previous download, or may be transmitted via an

35    intermediary. FIGURE 7 is illustrative of a window 93 produced by a WWW

browser 70 presenting an electronic form acquired from the Internet 20. The form generally includes one or more fields of information 91, which the user is required to complete and submit to the content provider to gain access to additional Web sites or to initiate a service being provided to user.

5      With continued reference to FIGURE 6, at block 604 the user activates a URL link 72 (FIGURE 4) to the fill server 34 causing a fill server application window to be generated. The fill server application window allows the user to obtain the information necessary to complete the electronic form in accordance with the present invention. It will be appreciated that the WWW browser 70 on the document
10 browser computer 30 preferably includes some feature that allows a user to save a preferred fill server Web site URL for faster access. Often, the link may be provided as a graphical drop down menu allowing a user to click on a textual/graphical representation of the link causing the URL to be automatically entered into the WWW browser 70.

15      Regardless of how the link is activated, the user transmits at block 606 a user ID and password to the fill server 34 via an input screen and login sequence illustrated in FIGURE 8 and described in detail below. Alternatively, as will be explained below, if the user has previously logged in to the fill server 34, the user ID and password may be automatically transferred to the fill server 34. With continued
20 reference to block 606, the document browser computer 30 also transmits a form signature of the electronic form to be completed. A form signature is a text string which can include data such as the URL where the electronic form is found, the label descriptors of one or more of the form's fields and a descriptor of the action requested by the form provider. The form signature is utilized by the fill server to create a form
25 map for the creation of a fill bundle corresponding to the form.

     FIGURE 8 is illustrative of a login/registration routine 800 implemented by user registration and login application 90 of the fill server 34 to process the user ID and password transmitted by the document browser computer 30 at block 608 of FIGURE 6. Routine 800 begins at block 802 with the display of a login/registration
30 screen on the user's display. As implemented in a graphical operating environment, an application window housing the login/registration screen is displayed in the fill server application window generated at block 604 (FIGURE 6). FIGURE 9 is illustrative of a login/registration screen 95 generated by the WWW browser 70 on the document browser computer 30. In this embodiment, the login/registration
35 screen 95 includes a registered user login portion 94 having a field for the user to

enter a user ID 96 and a field to enter a password 98. The login portion 94 allows a user to access their specific user profile from any WWW browser having access to the fill server URL. The login/registration screen 95 also includes a new registration portion 92 which allows a user to utilize create a new user profile.

5      With continued reference to FIGURE 8, at decision block 804, a determination is made whether the user wishes to create a new user profile or access an existing user profile. With reference to FIGURE 9, such a determination can be made by whether the user enters an ID and a password at the login portion 94, or whether the user clicks on the new registration portion 92. If at block 804 the user

10    wishes to create a new user profile, a new account registration icon 92 (FIGURE 9) is selected at block 806 and the fill sever application window on the WWW browser 70 displays a profile creation screen 99 at block 808.

FIGURES. 10A and 10B are illustrative of a profile creation screen 99 (block 808) displayed by the WWW browser 70 on the document browser

15    computer 30 for completing an initial user profile. With reference to FIGURE 10A, an illustrative profile creation screen 99 includes a portion 100 for the user to enter his or her initial contact information such as first and last name 102, a contact e-mail address 104, a user ID 106 and a password 108. In an actual embodiment, the first portion 100 entails the minimum amount of information required to be entered by the

20    user to have a registration. As would be readily understood by one skilled in the art, the amount and type of data required in the initial user registration would vary dependent on the type of user and the services intended purpose.

In a second portion 110 of the profile creation screen 99, the user indicates the level of security the user wishes to assign his or her profile information. In one

25    actual embodiment, the user is presented with a choice of a limited, moderate or extensive privacy guard of the user information. The indication of one of these three levels will be then used by the fill server 34 to determine whether a form field will be automatically completed dependent on the proposed use by the form vendor. For example, some proposed uses of profile information include for personalization of a

30    Web site, for completion of an activity such as an e-commerce transaction or for system administration. If a user selects an extensive security level, the fill server 34 would be prevented from using the user's profile to complete fields for Web site personalization, but could use the profile for completing fields required to effect an electronic transaction. As would be readily understood, various security

configurations and/or intended uses could be incorporated into the registration process.

                  With reference to FIGURE 10B, in a third portion 112 of the profile creation screen 99, the user is prompted for additional information that allows the fill

5      server 34 to further develop the user profile. Such information can include the user's gender and whether the fill server service is being used for work or home use. As would be readily understood, additional or modified fields in the third portion 112 would be considered to be within the scope of the present invention. Moreover, in an alternative embodiment, the third portion 112 may be omitted completely without

10     changing the function of the profile creation window.

                  With continued reference to FIGURE 10B, the profile creation screen 99 includes a fourth portion 114 prompting the user to save the fill server URL into their WWW browser 70. In one actual embodiment, the user is required to manually save the URL via their standard browser procedure. In an alternative embodiment, this

15     function may be performed automatically in the profile creation process. The initial profiled creation screen 99 also includes a submit/cancel button 116 allowing a user to send the initial registration data to the fill server 34 upon its completion or to cancel the registration process.

                  Returning to FIGURE 8, at block 810, the user enters the required data and

20     the entered information is posted to the fill server 34 at block 812. At block 814, the fill server 34 receives the new registration information and establishes a user profile. With reference to FIGURE 2, the fill server 34 communicates the profile registration information to the user information server 40 which stores the information under the user ID in the user information database 42.

25             Having established a new profile, at block 816, the user is given the opportunity to complete additional fields of data within the profile that may be utilized to complete electronic forms. FIGURE 11 is illustrative of a window 117 displayed by a WWW browser 70 on a document browser computer 30 and generated by a fill server 34 for modifying and/or expanding a user's profile. In this

30     window 117, the user is presented with additional grouping of fields 118 to complete, such as a full name field 120, a birth date field 122, a mobile phone field 124 and a pager/voice mail field 126. By scanning through the information, the user has the opportunity to select which of these fields will be completed.

                  In an alternative embodiment illustrated in FIGURE 12, a user may access a

35     separate fill server Web site at any point after the creation of a user profile for the

purpose of updating/expanding the profile. Similar to the login screen of FIGURE 9, the login window 127 includes a user ID and password 128 field and a new user registration field 130. Upon logging in, the user is presented with the update/modify window 127 of FIGURE 11 and completes the information in the manner described

5    above.

Returning to FIGURE 8, once the user has completed entering additional information at block 816, or if the user does not choose to enter in additional information, the fill server 34 provides a cookie to be deposited into the WWW browser 70 of the document browser computer 30 at block 824. As is generally

10   understood, a cookie is a user specific identifier containing state information assigned to the user whenever the user connects to a content provider's resource. The cookie is deposited into the WWW browser 70 and the WWW browser 70 is instructed to show the cookie's state information to the resource upon subsequent visits to the fill server 34.

15   Cookies generally facilitate the customization of Web sites for users and are vital to the implementation of multiple, composite Web sites. Some resources assign temporary cookies, which expire at the end of a session or when the browser application is terminated. Other cookies, however, are assigned for a longer duration facilitating identification beyond a single network session. The more permanent

20   cookies can be used to identify user-specific network history and to relay configuration preferences on a repeated basis. Generally, the transmittal of the information from the user computer system to the content provider is transparent to the user.

In one actual embodiment, at block 824, the fill server 34 deposits a

25   temporary cookie which will only be valid during the current browser session. The use of temporary cookies prevent a second user from accessing the first user's login data once the first user terminates the browser session. Alternatively, the fill server 34 may deposit more permanent cookies in the WWW browser 70 for continuous access to the user login information. Upon setting the cookie at

30   block 824, the routine 800 terminates at block 825.

Returning to decision block 804 of FIGURE 8, if a determination is made that the user wishes to access an existing user profile, the user enters his or her user ID and password at block 818. The information is posted to the fill server 34 at block 820, which will determine whether the login information is correct at decision

35   block 822. With reference to FIGURE 2, the fill server 34 polls the user information

server 40 to verify that the user ID and password match with one of the user profile accounts stored in the user information database 42. If at block 822 the login information is not correct, the routine returns to block 802 displaying the login/registration screen 95 (FIGURE 9). Alternatively, if at block 822 the login

5    information is correct, the fill server 34 provides a cookie to be deposited into the WWW browser 70 of the document browser computer 30 at block 824 and the routine 800 terminates at block 825.

Returning to FIGURE 6, after the user ID, password and form signature have been passed to the fill server at block 606, the document browser computer 30

10    obtains a representation of a fill bundle corresponding to the designated form from the fill server 34 at block 608. In an actual embodiment, the fill bundle includes a shippable code segment containing a merger of raw user data from a user profile and a form map corresponding to the electronic form to be completed. Additionally, the shippable code segment includes commands, which when merged into a WWW

15    browser application window, causes one or more fields to be completed. The generation of the fill bundle by the fill server 34 is described in more detail below in connection with FIGURE 15.

FIGURE 13 illustrates a window 110 displayed by the WWW browser 70 of the document browser computer 30 and generated by the fill server 34 in accordance

20    with the present invention. In the illustrated embodiment, the window 110 is very similar to the login/registration screen 95 of FIGURE 9 except that the user login portion 94 (FIGURE 9) has been replaced by an icon 111 labeled "My Information". In this embodiment, the icon 111 provides the user with the ability to access the fill bundle generated by the fill server 34 as will be described in more detail below.

25    With continued reference to FIGURE 6, once the Web browser 70 of the document browser computer 30 obtains and displays the fill bundle window 110 at block 608, the user transfers the fill bundle icon 111 from the fill bundle window 110 to the electronic form application window 93 by clicking on the icon 111 and "dragging" it to the electronic form application window 93. FIGURES 14A and 14B

30    are illustrative of the electronic form window 93 and fill bundle window 110 during the transfer of the fill bundle icon 111 to the electronic form window 93. Because of the general proximity of the two windows, the transfer of the icon 111 is accomplished by clicking on it and moving it to an area not covered by the fill bundle window 110, but covered by the electronic form application window 93. As would

be understood, the action of clicking and dragging an icon in a graphical windowed operating environment is generally understood.

In the illustrative embodiment, the fill bundle is embodied in a shippable code segment in the form of a JavaScript™ program. Shippable code segments may be generally understood to represent as computer-readable instructions which can be transferred from the fill server 34 and executed by the document browser computer 30. The dragging and depositing of a JavaScript™ program causes the receiving application window to merge the program with its existing code segments thereby allowing the code to be executed in the receiving application window. In this illustration, the JavaScript™ commands are used to electronically complete the fields of the form. It will be generally understood, that once incorporated into the target application window, the specific JavaScript™ commands utilized by a WWW browser, such as WWW browser 70 of the document browser computer 30, to electronically complete a form are known. It should be further understood that because the fill bundle is transferred as shippable code, the completion of the electronic form takes place on the document browser computer 30, but does not require the document browser computer 30 to install or maintain additional software. The utilization of other forms of shippable code is considered to be within the scope of the present invention.

FIGURE 14C is illustrative of the electronic form window 93 of FIGURE 7 after the fields have been completed in accordance with the present invention. In this illustration, the executed fill bundle has completed the first name 132, last name 134 and e-mail address 136 fields of the form automatically.

With continued reference to FIGURE 6, at block 612, the user completes any additional fields within the electronic form that the fill server 34 was unable to complete. Additionally, the user may correct or delete any of the data in the fields that were completed by the execution of the fill bundle.

Upon completion of the fields at block 612, or if the execution of the fill bundle has completed the necessary fields at block 610, the document browser computer 30 transfers the completed electronic form to the document server 32 at block 614. This is typically accomplished by the user indicating that he or she wishes to send the form by clicking on a "submit" or "send" button (or something similar).

In one actual embodiment, at block 614, the document browser computer 30 also sends a copy of the completed electronic form to the fill server 34. In this

embodiment, the transfer can be transparent to the user or may require user approval. Accordingly, the fill server 34 accepts the data and verifies whether the user has entered data for fields not previously completed in the user's account, or whether the data stored on the user information database 42 needs to be updated. In this

5    embodiment, as the user utilizes the fill server service to partially complete an electronic form, any additional data that is competed by the user after the fill bundle has been executed is automatically added to the user profile. Accordingly, continuous use of the fill server service from any document browser computer 30 maintains the user profile and allows for its expansion. The routine 600 then

10   terminates at block 615.

FIGURE 15 is a flow diagram illustrative of a routine 1500 implemented by the fill bundle generator application 88 of the fill server 34 for generating a fill bundle in accordance with the present invention. The routine 1500 describes the steps taken by the fill server 34 once the document browser computer 30 obtains a

15   user cookie (either by a previous use or by the registration/login routine illustrated in FIGURE 8) and is ready to transfer the information to the fill server 34 to obtain a fill bundle icon 111(at block 608 of FIGURE 6). At block 1502, the fill bundle generator application 88 obtains the user specific cookie that identifies which user profile stored in the user information database 42 will be utilized to complete the

20   electronic form. Additionally, and as mentioned previously, the fill bundle generator application 88 obtains a document signature for the electronic form.

In the one actual embodiment, the form signature of the document can include the URL of the form, descriptors of some or all of the form fields and/or descriptors of some or all of the required actions by the user. As would be generally understood,

25   current Internet browser capabilities limit the amount of data transferred in any URL link to 2000 characters. Accordingly, in this embodiment, the present invention creates the signature of the electronic form such that it equals less than 2000 characters. However, as would be readily understood, the limitation of the number of characters allowed in a form signature may vary dependant on the capabilities of the

30   Internet browser.

Upon receipt of the user cookie and form signature, the fill bundle generator application 88 utilizes the user cookie to retrieve a corresponding user profile from the user information server 40 at block 1504. In this illustration, the fill bundle generator application 88 passes the user ID and password to the user information

35   server 40 which retrieves the user profile and returns it to the fill bundle generator

application 88. Additionally, and essentially parallel, at block 1506, the fill bundle generator application 88 utilizes the electronic form signature to obtain a form map of the fields of the electronic form (e.g., from the form map database 38). A form map includes the association of the specific description of the fields on the form with

5     a standard description of the user data. For example, a specific form requiring a user name may label a field requesting a user's first name as "First Name". Additionally, the user profile data corresponding to this field is given the standard label as "user_name_first" in the user information database 42. Accordingly, the form map would associate the field "FirstName" with its standard field "user_name_first"

10    allowing the fill bundle generator application 88 to return the appropriate user profile data. In this embodiment, the association of the specific description of the fields on the form with the standard description is referred to as a tuple.

In one actual embodiment, there are several methods of obtaining the form map. In a first method, the fill bundle generator application 88 passes the form

15    signature to the form map server 36 which determines whether a form map for the electronic form already exists. If so, the form map is retrieved from the form map database 38 and transferred to the fill bundle generator application 88. The form maps stored in the form map database may be created by having a technician obtain and manually key in each form tuple. Although potentially time consuming, this

20    approach increases the correctness of the form tuple association.

Often, however, electronic forms are modified on a more frequent basis. The modifications may include minor changes to the form, such how the fields are ordered or the presentation of the fields on the display. However, some modifications, such as the changing of the URL where the form is located or

25    removing one or more fields, may alter the form sufficiently to render a previous form map incorrect or unrecognizable. In the event that the signature of an electronic form is modified sufficiently, or a manually typed form map has not been created, the fill bundle generator application 88 creates a new form map utilizing a form map generator. In this illustration, the form map generator obtains the descriptors of the

30    form fields from the form signature and attempts to match the field descriptor with a standard field descriptor in an iterative manner. If the field descriptors do not match identically, the form map generator may make a "best guess" association utilizing factors such as partially matching field descriptor names, previous matching history and/or user preferences for best guess associations. An illustrative form map

35    generating method and structure is described in commonly owned U.S.

Nonprovisional Patent Application Serial No. 09/357,530 entitled INTELLIGENT MAPPING OF FIELD NAMES IN AN ELECTRONIC FORM WITH STANDARD FIELD NAMES, and filed July 19, 1999, which is specifically incorporated herein by reference. As would be readily understood by someone skilled in the relevant art,

5      alternative field mapping methods and structures would be considered to be within the scope of the present invention. The resulting form map may be stored in the form map database 38 for future use.

Having obtained the form map at 1506 and the raw user data at block 1504, the fill bundle generator application 88 merges the form map and the raw user data at

10     block 1508. FIGURE 16 is a flow diagram illustrative of a sub-routine 1600 implemented by the fill bundle generator application 88 for merging a electronic form map and raw user data in accordance with the present invention. Routine 1600 begins with the fill bundle generator application 88 obtaining a tuple associating a form field descriptor and a standard user profile descriptor at block 1602. At

15     block 1604, a corresponding user profile raw data component is retrieved. The user profile raw data component can include a field descriptor label, a data string containing the actual user profile data and a security preference associated with the data. At block 1606, the fill bundle generator application 88 compares the tuple from the form map with the raw data component and determines whether the security

20     preference allows for the completion of the field at decision block 1608. If the user profile indicates that the corresponding field cannot be used for the requested purpose, the match will not be accepted and the user will be informed at block 1610 that data requested for that field was to be utilized for an unauthorized purpose. In one actual embodiment, if a form tuple is rejected, the fill bundle generator

25     application 88 terminates the merger routine 1600 at block 1615 and a fill bundle is not generated. Alternatively, a fill bundle may be generated having only the accepted tuple mergers completed, giving the user the option of completing the rejected fields as described at block 612 (FIGURE 6).

If at block 1608 the requested user data is authorized, the raw data string is

30     merged with the requested field form map at block 1612. In one actual embodiment, the merged data is maintained at the fill bundle generator application 88 to be incorporated into a JavaScript™ code segment at the completion of the merging routine. Accordingly, at decision block 1614, a determination is made as to whether there are any tuples remaining in the form map. If so, the sub-routine 1600 returns to

block 1608 for continued processing. If there are no further tuples at block 1618, the routine is terminated at block 1615.

With reference to FIGURE 15, once the form map and the user profile raw data has been merged, the data is converted into shippable code for transmittal to the document browser computer 30 at block 1510. In one actual embodiment, the merged data is embodied in shippable code in the form of JavaScript™ commands for entering the data string into the corresponding field of the electronic form. In this actual embodiment, the shippable code is displayed graphically as the fill bundle icon 111 on the WWW browser 70 of the document browser computer 30. Accordingly, at block 1512, the fill bundle is transmitted to the document browser via the Internet 20. In the illustrative embodiment, the transfer of the user data is facilitated through the use of a Secured Socket Layer communication protocol to ensure user data security. As would be readily understood, additional or alternative security features, such as private dedicated network links or alternative security protocols could be incorporated into the system. At this point, the fill bundle generation routine 1500 is complete at block 1514 and the completion of the one or more fields of the electronic form occur without requiring further action from the fill server 34.

FIGURE 17 is a flow diagram illustrative of a sub-routine 1700 implemented by the fill bundle generator application 88 of the fill server 34 for updating a user profile in accordance with the present invention. The sub-routine 1700 begins at block 1702 with the fill bundle generator application 88 obtaining a completed electronic form at in a manner described at block 614 of FIGURE 6. At block 1704, the fill bundle generator application 88 attempts to locate a complete form map from the form map database 38. If the corresponding form map does not exist, has been deleted, or is incomplete, at block 1706, the fill bundle generator application 88 generates a new form map in a manner described above. If the form map exists at block 1704 or after a form map is created at block 1706, the raw user data corresponding to an electronic form field is parsed from completed form at block 1708.

At decision block 1710, a determination is made whether the user data parsed from the completed form matches the user data stored in the user profile. If it does not match, or if the user profile does not contain data for the identified field, the user profile data is updated at block 1712. At decision block 1714, a determination is made whether there exists any additional fields on the form to be parsed and

inspected. If so, the process returns to block 1710. If, however, no further fields need to be parsed and inspected, the routine is completed at block 1716. By updating the user profile as electronic forms are submitted, the fill bundle generator application 88 maintains current user data and allows the user to expand the profile

5    by filling out more electronic forms. As would be readily understood, the updating sub-routine is independent from and not considered essential to the other aspects of the invention.

Although the present invention has been described in terms of utilizing shippable code which merges into the electronic form to be completed, it may be

10    possible to implement some or all of the functions of the shippable code by signing the code. In the alternative embodiment, the fill bundle would be transferred to the document browser computer 30 as signed JAVA™ code segments and would interact with form application windows to complete the fields. The creation and execution of signed JAVA™ applications is known to those skilled in the art. Nevertheless, from

15    a user perspective a fill bundle would still be selected and dragged to an electronic form to initiate the completion of one or fields. The use of signed applications would be considered to be within the scope of the present invention.

By utilizing shippable code segments which are presented to a user as a graphical/textual representation, the present invention facilities the completion of one

20    or more fields of an electronic form. Moreover, utilizing standard WWW Web browser functionality, the present invention facilities a user's access to personal information from any number of computer systems without requiring the user to install or maintain software applications and without requiring cross-communication between computer systems.

25    While the preferred embodiment of the invention has been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the invention.